Regularized maximum correntropy machine

Jim Jing-Yan Wang[a], Yunji Wang[b], Bing-Yi Jing[c], Xin Gao[a,*]

[a]*Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia*
[b]*Electrical and Computer Engineering Department, The University of Texas at San Antonio, San Antonio, TX 78249, USA*
[c]*Department of Mathematics, Hong Kong University of Science and Technology, Kowloon, Hong Kong*

Abstract

In this paper we investigate the usage of regularized correntropy framework for learning of classifiers from noisy labels. The class label predictors learned by minimizing transitional loss functions are sensitive to the noisy and outlying labels of training samples, because the transitional loss functions are equally applied to all the samples. To solve this problem, we propose to learn the class label predictors by maximizing the correntropy between the predicted labels and the true labels of the training samples, under the regularized Maximum Correntropy Criteria (MCC) framework. Moreover, we regularize the predictor parameter to control the complexity of the predictor. The learning problem is formulated by an objective function considering the parameter regularization and MCC simultaneously. By optimizing the objective function alternately, we develop a novel predictor learning algorithm. The experiments on two challenging pattern classification tasks show that it significantly outperforms the machines with transitional loss functions.

Keywords: Pattern classification, Label noise, Maximum Correntropy criteria, Regularization

[*] Correspondence should be addressed to Xin Gao. Tel: +966-12-8080323.

1. Introduction

The classification machine design has been a basic problem in the pattern recognition field. It tries to learn an effective predictor to map the feature vector of a sample to its class label [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. We study the supervised multi-class learning problem with L classes. Suppose we have a training set denoted as $\mathcal{D} = \{(x_i, y_i)\}, i = 1, \cdots, N$, where $x_i = [x_{i1}, \cdots, x_{iD}]^\top \in \mathbb{R}^D$ is the D dimensional feature vector of the i-th training sample, and $y_i \in \{1, \cdots, L\}$ is the class label of i-th training sample. Moreover, we also denote the label indicator matrix as $Y = [Y_{li}] \in \mathbb{R}^{L\times N}$, and $Y_{li} = 1$ if $y_i = l$, and -1 otherwise. We try to learn L class label predictors $\{f^l_\theta(x)\}, l = 1, \cdots, L$ for the multi-class learning problem, where $f^l_\theta(x)$ is the predictor for the l-th class and θ is its parameter. Given a sample x_i, the output of the l-th predictor is denoted as $f^l_\theta(x_i)$, and we further denote the prediction result matrix as $F_\theta = [F_{\theta li}] \in \mathbb{R}^{L\times N}$, and $F_{\theta li} = f^l_\theta(x_i)$. To make the prediction as precise as possible, the target of predictor learning is to learn parameter θ, so that the difference between true class labels of the training samples in Y and the prediction results in F_θ could be minimized, while keeping the complexity of the predictor as low as possible. To measure how well the prediction results fit the true class label indicator, several loss functions $L(F_\theta, Y)$ could be considered to compare the prediction results in F_θ against the true class labels of the training samples in Y, such as the 0-1 loss function, the square loss function, the hinge loss function, and the logistic loss function. We summarize various loss functions in Table 1.

These loss functions introduced in Table 1 have been used widely in various learning problems. One common feature of these loss function is that a sample-wise loss function is applied to each training sample equally and then the losses of all the samples are summed up to obtain the final overall loss. The sample-wise loss functions are of exactly the same form with the same parameter (if they have parameters). The basic assumption behind this loss function is that the training samples are of the same importance. However, due to the limitation of the sampling technology and noises occurred during the sampling procedure,

Table 1: Various empirical loss functions for predictor learning

Title	Formula of $L(F_\theta, Y)$	Notes
0-1 Loss	$\sum_{i,l} \mathbb{I}[F_{\theta li} Y_{li} < 0]$, where $\mathbb{I}(\cdot)$ is the indicator function and $\mathbb{I}(\cdot) = 1$ if $(\cdot)$ is true, 0 otherwise.	The 0-1 loss function is NP-hard to optimize, non-smooth and non-convex.
Square Loss	$\sum_{i,l} [F_{\theta li} - Y_{li}]^2 = \|\|F_\theta - Y\|\|^2$, where $\circ$ denotes the element wise product of two matrices, and $\mathbf{1}_{N\times L}$ is a $N\times L$ matrix with all elements of ones.	The square loss function is a convex upper bound on the 0-1 loss. It is smooth and convex, thus easy to optimize.
Hinge Loss	$\sum_{i,l}[1 - F_{\theta li} Y_{li}]_+ = \mathbf{1}_N^\top [\mathbf{1}_{N\times L} - F_\theta \circ Y]_+ \mathbf{1}_L$ where $[x]_+ = max(0, x)$, and $\mathbf{1}_N \in \mathbb{R}^N$ is a column vector with all ones.	The hinge loss function is not smooth but subgradient descent can be used to optimize it. It is the most common loss function in SVM.
Logistic Loss	$\sum_{i,l} ln[1 + e^{-F_{\theta li} Y_{li}}] = \mathbf{1}_N^\top ln \left[\mathbf{1}_{N\times L} + e^{-F_\theta \circ Y}\right] \mathbf{1}_L$	This loss function is also smooth and convex, and is usually used in regression problem.

there are some noisy and outlying samples in real-world applications. If we use the transitional loss functions listed in Table 1, the noisy and outlying training samples will play more important roles even than the good samples. Thus the predictors learned by minimizing the transitional loss functions are not robust to the noisy and outlying training samples, and could bring a high error rate when applied to the prediction of test samples.

Recently, regularized correntropy framework has been proposed for robust pattern recognition problems [11, 12, 13, 14]. In [15], He et, al argued that the classical mean square error (MSE) criterion is sensitive to outliers, and introduced the correntropy to improve the robustness of the presentation. Moreover, the l_1 regularization scheme is imposed on the correntropy to learn robust and sparse representations. Inspired by their work, we propose to use the regularized correntropy as a criterion to compare the prediction results and the true class labels. We use correntropy to compare the predicted labels and the true labels, instead of comparing the feature of test sample and its reconstruction from the training samples in He et, al's work. Moreover, an l_2 norm regularization is introduced to control the complexity of the predictor. In this way, the predictor learned by maximizing the correntropy between prediction results and the true labels will be robust to the noisy and outlying training samples. The proposed classification Machine Maximizing the Regularized CorrEntropy, which is called RegMaxCEM, is supposed to be more insensitive to outlining samples than the ones with transitional loss functions. Yang et, al. [16] also proposed to use correntropy to compare predicted class labels and true labels. However, in their framework, the target is to learn the class labels of the unlabeled samples in a transductive semi-supervised manner, while we try to learn the parameters for the class label predictor in a supervised manner.

The rest of this paper is structured as follows: In Section 2, we propose the regularized maximum correntropy machine by constructing an objective function based on the maximum correntropy criterion (MCC) and developing an expectation – maximization (EM) based alternative algorithm for its optimization. In Section 3, the proposed methods are validated by conducting extensive experiments on two challenging pattern classification tasks. Finally, we give the conclusion in Section 4.

2. Regularized Maximum Correntropy Machine

In this section we will introduce the classification machine maximizing the correntropy between the predicted class labels and the true class labels, while keeping the solution as simple as possible.

2.1. Objective Function

To design the predictors $f_\theta^l(x)$, we first represent the data sample x as $\widetilde{x}$ in the linear space and the kernel space as:

$$\widetilde{x} = \begin{cases} x, & (linear), \\ K(\cdot, x), & (kernel), \end{cases} \tag{1}$$

where $K(\cdot, x) = [K(x_1, x), \cdots, K(x_N, x)]^\top \in \mathbb{R}^N$ and $K(x_i, x_j)$ is a kernel function between x_i and x_j. Then a linear predictor $f_\theta^l(x)$ will be designed to predict whether the sample belongs to the l-th class as

$$f_\theta^l(x) = w_l^\top \widetilde{x} + b_l, \ l = 1, \cdots, L, \tag{2}$$

where $\theta = \{(w_l, b_l)\}_{l=1}^L$ is the parameters of the predictors, $w_l \in \mathbb{R}^D$ is the linear coefficient vector and $b_l \in \mathbb{R}$ is a bias term for the l-th predictor. The target of predictor designing is to find the optimal parameters to have the prediction result $f_\theta^l(x_i)$ of the i-th sample to fit its true class label indicator Y_{li} as well as possible, while keeping the solution as simple as possible. To this end, we consider the following two problems simultaneously when designing the objective function:

Prediction Accuracy Criterion based on Correntropy To consider the prediction accuracy, we could learn the predictor parameters by minimizing a loss function listed in Table 1 as

$$\underset{\theta}{min} L(F_\theta, Y) \tag{3}$$

However, as we mentioned in Section 1, all these loss functions are applied to all the training samples equally, which is not robust to the noisy samples and outlying samples. To handle this problem, instead of minimizing a loss function to learn the predictor, we use the MCC [11] framework to learn the predictor by maximizing the correntropy between the predicted results and the true labels.

Remark 1: In previous studies, it has been claimed that the MCC is insensitive to outliers. For example, in [11], it is claimed that "the maximum correntropy criterion, ... is much more insensitive to outliers." Based on this fact, we assume that the predictors developed based on MCC should also be insensitive to outliers.

Correntropy is a generalized similarity measure between two arbitrary random variables A and B. However, the joint probability density function of A and B is usually unknown, and only a finite number of samples of them are available as $\{(a_i, b_i)\}_{i=1}^{d}$. It leads to the following sample estimator of correntropy:

$$V(A,B) = \frac{1}{d}\sum_{i=1}^{d} g_\sigma(a_i - b_i), \tag{4}$$

where $g_\sigma(a_i - b_i) = exp\left(-\frac{(a_i-b_i)^2}{2\sigma^2}\right)$ is a Gaussian kernel function, and σ is a kernel width parameter. For a learning system, MCC is defined as

$$max_\vartheta \frac{1}{d}\sum_{i=1}^{d} g_\sigma(a_i - b_i) \tag{5}$$

where ϑ is the parameter to be optimized in the criterion so that B is as correlated to A as possible.

Remark 2: ϑ is usually a parameter to define B, but not the kernel function parameter σ. In the learning system, we try to learn ϑ so that with the learned ϑ, B is correlated to A. For example, in this case, A is the true class label matrix while B is the predicted class label matrix, and ϑ is the predictor parameter to define B.

To adapt the MCC framework to the predictor learning problem, we let A be the prediction result matrix F_θ parameterized by θ, and B be the true class label matrix Y, and we want to find the predictor parameter θ such that F_θ becomes as correlated to Y as possible under the MCC framework. Then, the following correntropy-based predictor learning model will be obtained:

$$\begin{aligned} &\underset{\theta}{max} V(F_\theta, Y), \\ &V(F_\theta, Y) = \frac{1}{L \times N} \sum_{l=1}^{L} \sum_{i=1}^{N} g_\sigma(F_{\theta li} - Y_{li}) \end{aligned} \tag{6}$$

Please notice that in [11], MCC is used to measure the similarity between a test sample and its sparse linear representation of training samples, while in this work it is used to measure the similarity between the predicted class label and its true label. Also note that the dependence on σ in (6) and later (8), (11) relies on the dependence of the kernel function $g_\sigma(\cdot)$. In our experiments, the σ value is calculated as $\sigma = \frac{1}{2\times L\times N} \sum_{l=1}^{L} \sum_{i=1}^{N} \|F_{\theta li} - Y_{li}\|_2^2$ following [11].

Predictor Regularization To control the complexity of the l-th predictor independently, we introduce the l_2-based regularizer $||w_l||^2$ to the coefficient vector w_l of the l-th predictor. We assume that the predictors of different classes are equally important, and the following regularizer is introduced for multi-class learning problem:

$$\underset{\{w_l\}_{l=1}^{L}}{min} \frac{1}{L} \sum_{l=1}^{L} ||w_l||^2 \tag{7}$$

Remark 3:The l_2 norm is also used by support vector regression as a measure of model complexity. However, in support vector classification, this regularization term is either obtained by a "maximal margin" regularization or obtained by a "maximal robustness" regularization for certain type of feature noises [17]. Thus our l_2 norm regularization term can also be regarded as a term to seek maximal margin or robustness.

Remark 4: The l_2-regularization is used in comparison to the l_1-regularization in our model. Using l_1-regularization we can seek the sparsity of the predictor coefficient vector, but it cannot guarantee the minimal model complexity, maximal margin or maximal robustness like the l_2-regularization, thus we choose to use the l_2-regularization. In the future, we will explore the usage of l_1-regularization to see if the prediction results can be improved.

By substituting $\theta = \{w_l, b_l\}_{l=1}^{L}$, $F_{\theta li} = f_{w_l,b_l}^{l}(x_i)$, and combining both the predictor regularization term in (7) and the prediction accuracy criterion term based on correntropy in (6), we obtain the following maximization problem for the maximum correntropy machine:

$$\underset{\{(w_l,b_l)\}_{l=1}^{L}}{max} \frac{1}{L \times N} \sum_{l=1}^{L} \sum_{i=1}^{N} g_\sigma(f_{w_l,b_l}^{l}(x_i) - Y_{li}) - \alpha \frac{1}{L} \sum_{l=1}^{L} ||w_l||^2 \tag{8}$$

where α is a tradeoff parameter. This optimization problem is based on correntropy using a Gaussian kernel function $g_\sigma(x)$. It treats the prediction of individual training samples of individual classes differently. By this way, we can give more emphasis on samples with correctly predicted class labels, while those noisy or outlying training samples will have small contributions to the correntropy. In fact, when the regularizer term is introduced, (8) is a case of the regularized correntropy framework [15].

2.2. Optimization

Due to the nonlinear attribute of the kernel function $g_\sigma(x)$ in the objective function in (8), direct optimization is difficult. An attribute of the kernel function $g_\sigma(x)$ is that its derivative is also the same kernel function, and if we set its derivative to zero to seek the optimization of the objective, it is not easy to obtain a close form solution. However, according to the property of the convex conjugate function, we have:

Proposition 1 There exists a convex conjugate function φ of $g_\sigma(x)$ such that

$$g_\sigma(x) = max_p(p||x||^2 - \varphi(p)) \tag{9}$$

and for a fixed x, the maximum is reached at $p = -g_\sigma(x)$. This Proposition is taken from [18], which is further derived from the theory of convex conjugated functions. It is further discussed and used in many applications such as [11, 15, 19, 20].

By substituting (9) to (8), we have the augmented optimization problem in an enlarged parameter space

$$\begin{aligned}&\underset{\{(w_l,b_l)\}_{l=1}^L,P}{max}\frac{1}{L\times N}\sum_{l=1}^{L}\sum_{i=1}^{N}\left[P_{li}||f_{w_l,b_l}^l(x_i)-Y_{li}||^2-\varphi(P_{li})\right]-\alpha\frac{1}{L}\sum_{l=1}^{L}||w_l||^2\\&=\frac{1}{L\times N}\sum_{l=1}^{L}\sum_{i=1}^{N}\left[P_{li}||w_l^\top\widetilde{x}_i+b_l-Y_{li}||^2-\varphi(P_{li})\right]-\alpha\frac{1}{L}\sum_{l=1}^{L}||w_l||^2,\end{aligned} \tag{10}$$

where $P=[P_{li}]\in\mathbb{R}^{N\times L}$ are the auxiliary variable matrix. To optimize (10), we adapt the EM framework to solve P and $\{(w_l,b_l)\}_{l=1}^L$ alternately.

2.2.1. Expectation Step

In the expectation step of the EM algorithm, we calculated the auxiliary variable matrix P by fixing θ. Obviously, according to **Proposition 1**, the maximum of (10) can be reached at

$$\begin{aligned}P&=-g_\sigma(F_\theta-Y),\\P_{li}&=-g_\sigma(w_l^\top\widetilde{x}_i+b_l-Y_{li}).\end{aligned} \tag{11}$$

Note that $g_\sigma(X)$ is the element-wise Gaussian function. With fixed predictor parameters, the auxiliary variable $-P_{li}$ can be regarded as confidence of prediction result of the i-th training sample regarding to the l-th class. The better the l-th prediction result of the i-th sample fits the true label Y_{li}, the larger the $-P_{li}$ will be.

Remark 5: It is interesting to see if there is any relation between the auxiliary variables in P and the slack variables in SVM. Actually, both the auxiliary variables in P and the slack variables in SVM can be viewed as measures of classification losses. The slack variables in SVM are the upper boundaries of hinge losses of the training samples, while the auxiliary variables in P are a dissimilarity measure between the predicted labels and the true labels under the framework of the MCC rule, which is also a loss function. Meanwhile, the auxiliary variables in P also play a role of weights of different training samples as in (10), so that the learning can be robust to the noisy labels, but the auxiliary variables in SVM do not have such functions.

Remark 6: In the expectation step, we actually solve an alternative optimization of solving P while fixing $\{(w_l, b_l)\}_{l=1}^L$. However, according to **Proposition 1**, the solution for this optimization problem is in the form of (11), which can be calculated directly and makes it an expectation step of the EM algorithm.

2.2.2. Maximization Step

In the maximization step of the EM algorithm, we solve the predictor parameters $\{(w_l, b_l)\}_{l=1}^L$ while fixing P. The optimization problem in (10) turns to

$$\underset{\{(w_l,b_l)\}_{l=1}^L}{max} \frac{1}{L \times N} \sum_{l=1}^{L} \sum_{i=1}^{N} \left[P_{li} ||w_l^\top \widetilde{x}_i + b_l - Y_{li}||^2 - \varphi(P_{li}) \right] - \alpha \frac{1}{L} \sum_{l=1}^{L} ||w_l||^2. \tag{12}$$

Noticing $P_{li} < 0$ and removing terms irrelevant to w_l and b_l, the maximization problem in (12) can be reformulated as the following dual minimization problem:

$$\begin{aligned} &\underset{\{(w_l,b_l)\}_{l=1}^L}{min} O(w_1, b_1, \cdots, w_L, b_L), \\ &O(w_1, b_1, \cdots, w_L, b_L) = \frac{1}{L \times N} \sum_{l=1}^{L} \sum_{i=1}^{N} (-P_{li} ||w_l^\top \widetilde{x}_i + b_l - Y_{li}||^2) + \alpha \sum_{l=1}^{L} ||w_l||^2. \end{aligned} \tag{13}$$

To simplify the notations, we define a vector $u_l = [u_{l1}, \cdots, u_{lN}]^\top \in \mathbb{R}^N$ so that $u_{li}^2 = -\frac{1}{N} P_{li}$. With u_l, the objective function in (13) can be rewritten as

$$\begin{aligned} O(w_1, b_1, \cdots, w_L, b_L) =& \frac{1}{L} \sum_{l=1}^{L} \left[||u_{li}(w_l^\top \widetilde{x}_i + b_l - Y_{li})||^2 + \alpha ||w_l||^2 \right] \\ =& \frac{1}{L} \sum_{l=1}^{L} \left[(w_l^\top \overline{X}_l + b_l u_l^\top - \overline{Y}_l)(w_l^\top \overline{X}_l + b_l u_l^\top - \overline{Y}_l)^\top + \alpha w_l^\top w_l \right], \end{aligned} \tag{14}$$

where $\overline{X}_l = [u_{l1}\widetilde{x}_1, \cdots, u_{lN}\widetilde{x}_N] \in \mathbb{R}^{D \times N}$ is the matrix containing all the training sample feature vectors weighted by u_l, and $\overline{Y}_l = [u_{l1}Y_{l1}, \cdots, u_{lN}Y_{lN}] \in \mathbb{R}^N$ is the l-th row of Y weighted by u_l.

Obviously, the optimization problem in (13) is a linear least squares problem. Analytical solution for Problem (13) could be obtained easily. By setting the derivative of $O(w_1, b_1, \cdots, w_L, b_L)$ with regard to b_l to zero, we have

$$\begin{aligned} &\frac{\partial O(w_1, b_1, \cdots, w_L, b_L)}{\partial b_l} = \frac{1}{2L}(w_l^\top \overline{X}_l + b_l u_l^\top - \overline{Y}_l)\mathbf{1}_N = 0 \\ &\Rightarrow b_l = \frac{(\overline{Y}_l - w_l^\top \overline{X}_l)\mathbf{1}_N}{u_l^\top \mathbf{1}_N} = \overline{y}_l - w_l^\top \overline{x}_l, \end{aligned} \tag{15}$$

where $\overline{y}_l = \frac{\overline{Y}_l \mathbf{1}_N}{u_l^\top \mathbf{1}_N}$ and $\overline{x}_l = \frac{\overline{X}_l \mathbf{1}_N}{u_l^\top \mathbf{1}_N}$. By substituting (15) to $O(w_1, b_1, \cdots, w_L, b_L)$, we have

$$O(w_1, \cdots, w_L) = \frac{1}{L}\sum_{l=1}^{L}\left\{[w_l^\top(\overline{X}_l - \overline{x}_l u_l^\top) - (\overline{Y}_l - \overline{y}_l u_l^\top)][w_l^\top(\overline{X}_l - \overline{x}_l u_l^\top) - (\overline{Y}_l - \overline{y}_l u_l^\top)]^\top + \alpha w_l^\top w_l\right\} \tag{16}$$

By setting the derivative of $O(w_1, \cdots, w_L)$ with regard to w_l to zero, we have the optimal solution w_l^*

$$\begin{aligned} &\frac{\partial O(w_1, \cdots, w_L)}{\partial w_l} = \frac{1}{2L}[2(\overline{X}_l - \overline{x}_l u_l^\top)(\overline{X}_l - \overline{x}_l u_l^\top)^\top w_l - 2(\overline{X}_l - \overline{x}_l u_l^\top)(\overline{Y}_l - \overline{y}_l u_l^\top)^\top + 2\alpha w_l] = 0 \\ &\Rightarrow w_l^* = [(\overline{X}_l - \overline{x}_l u_l^\top)(\overline{X}_l - \overline{x}_l u_l^\top)^\top + \alpha I]^{-1}(\overline{X}_l - \overline{x}_l u_l^\top)(\overline{Y}_l - \overline{y}_l u_l^\top)^\top, \end{aligned} \tag{17}$$

where I is an $D \times D$ identity matrix. Then we substitute w_l^* to (15), and we will have the optimal solution of b_l^*,

$$b_l^* = \overline{y}_l - {w_l^*}^\top \overline{x}_l \tag{18}$$

2.3. Algorithm

Algorithm 1 summarizes the predictor parameter learning procedure of RegMaxCEM. The E-step and the M-step will be repeated for T times.

Algorithm 1 RegMaxCEM Learning Algorithm.

```
Input: Training set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$;
Initialize the auxiliary variable matrix $P^0 = -\mathbf{1}_{L\times N}$;
Represent each sample $x_i$ as $\widetilde{x}_i$ as in (1);
for $t = 1, \cdots, T$ do
    Maximization-Step: Update the predictor parameters $\theta^t = \{(w_l^t, b_l^t)\}_{l=1}^L$
    as in (17) and (18) by fixing $P^{t-1}$.
    Expectation-Step: Update the auxiliary variable matrix $P^t$ as in (11)
    by fixing the predictor parameters $\theta^t$.
end for
Output: Predictor parameters $\theta^T = \{(w_l^T, b_l^T)\}_{l=1}^L$.
```

3. Experiments

In the experiments, we will evaluate the proposed classification method on two challenging pattern classification tasks — bacteria identification [21] and prediction of DNA-binding sites in proteins [22].

3.1. Experiment I: Bacteria Identification

3.1.1. Dataset and Setup

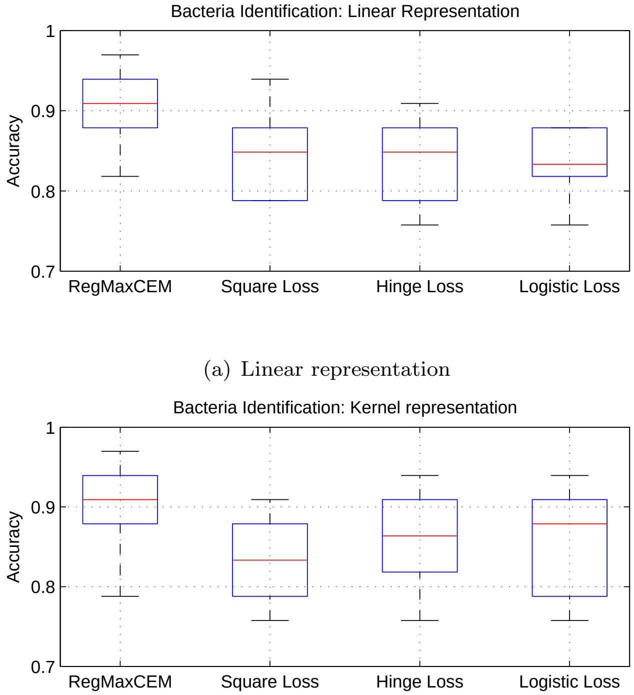
High-precision identification of bacteria is quite important for the diagnosis of cancers and bacterial infections. Recently, ensemble aptamers (ENSaptamers), which utilizes a small set of nonspecific DNA sequences, has been proposed to provide an effective platform for the detection of bacteria [21]. ENSaptamers is a sensor array with seven sensors, and each sensor is designed using a DNA element.

For the experiment, we collected in total 66 samples of 6 different bacteria, including S.tyohimurium, S.flexneri, E.coli (CAU 0111), S.sonnei, S.typhi and E.coli (ATCC 25922). The number of samples for each bacteria varies from 9 to 13. Given an unknown bacteria sample with its fluorescence response patterns of ENSaptamer, the task is to identify which bacteria it is. To this end the seven fluorescence response patterns of ENSaptamer against the sample will be used to construct the 7-dimensional feature vector, and then the sample will be classified into one of the known bacteria using the RegMaxCEM predictor.

To conduct the experiment, we randomly split the entire dataset into two non-overlapping subsets — the training set and the test set. 33 samples were used as training sample in the training set, while the remaining 33 ones as test samples. The predictor parameters of RegMaxCEM were trained using the feature vectors and class labels of the training samples. Then the class labels of the test samples were predicted by the trained predictor, and compared to their true labels to calculate the classification accuracy. The random split process (training/test) was repeated for ten times and the accuracies over these ten splits were reported as classification performance.

3.1.2. Results

We compare our proposed method against other loss function based classifiers, including square loss, hinge loss and logistic loss. 0-1 loss is the simplest loss function, but difficult to optimize, thus is not compared in the experiment. The boxplots of accuracies of different methods using both linear and kernel representations are illuminated in Figure 1. As shown in Figure 1, predictor produced by maximizing the correntropy yields improvements over other loss functions. Given the extremely small variation of classification accuracies over the ten splits, though the improvement of the accuracies are not large in absolute terms (around 0.1), it is consistent and significant. To verify whether the improvements are statistically significant, we performed the paired t-tests to the accuracies of the proposed method and other compared methods. The null hypothesis of the T-test is that the accuracies of the proposed method and the compared methods come from distributions with equal means. The P values of the t-tests are reported as measurements of statistically significance. A low P value implies that the difference between the proposed method and the compared methods are statistically significant. The P values are reported in Table 2. As we can see from the table, all the improvements archived by RegMaxCEM, for both linear representation and kernel representation, are statistically significant at the 0.05 significance level. This is not surprising: There are some noisy and outlying samples in the training set, which have been utilized by the methods with square loss, hinge loss or logistic loss as equally as other samples, thus they bring some bias to the predictor. However, the RegMaxCEM has the potential of filtering these samples, which can result in reliable learning of predictors in practice. It is also interesting to notice that the square loss, hinge loss and logistic loss have archived very similar classification accuracies. Though they used different loss functions, these loss functions are applied to the training samples equally.



Bacteria Identification: Linear Representation

(a) Linear representation

Bacteria Identification: Kernel representation

(b) Kernel representation

Figure 1: Boxplots of accuracies of bacteria identification.

Table 2: P values of paired T-tests on accuracies of ten splits of RegMaxCEM and compared methods on bacteria identification.

Linear representation		Kernel representation	
Compared methods	P values	Compared methods	P values
Square Loss	0.0266	Square Loss	0.0118
Hinge Loss	0.0243	Hinge Loss	0.0224
Logistic Loss	0.0115	Logistic Loss	0.0095

3.2. Experiment II: DNA-Binding Site Prediction

It is very important to predict the DNA-binding sites in proteins for understanding the molecular mechanisms of protein-DNA interaction. In this experiment, we will evaluate the proposed method for prediction of DNA-binding sites [22].

3.2.1. Dataset and Setup

The PDNA-62 database for DNA-binding site prediction has been used in this experiment. This database contains 8,163 sites in proteins in total. Among these sites, 1,215 of them are DNA-binding sites, while the remaining 6,948 sites are non-binding sites. We select 1,000 DNA-binding sites and 5,000 non-binding sites from the PDNA-62 database to construct our database for the experiment. Given a candidate site, the goal of DNA-binding site prediction is to predict whether it is a DNA-binding site or not. To this end, the evolutionary information, solvent accessible surface area and the protein backbone structure features were extracted from the site, and then combined to construct the feature vector. The feature vector was further inputted into the classifier to distinguish DNA-binding sites from the non-binding sites [22].

To conduct the experiment, we employed the 10-fold cross validation. The database was split into 10 non-overlapping folds randomly, one of which was used as the test set, while the rest 9 of them were used as the training set. The procedure was repeated for 10 times so that each fold was used as the test set once.

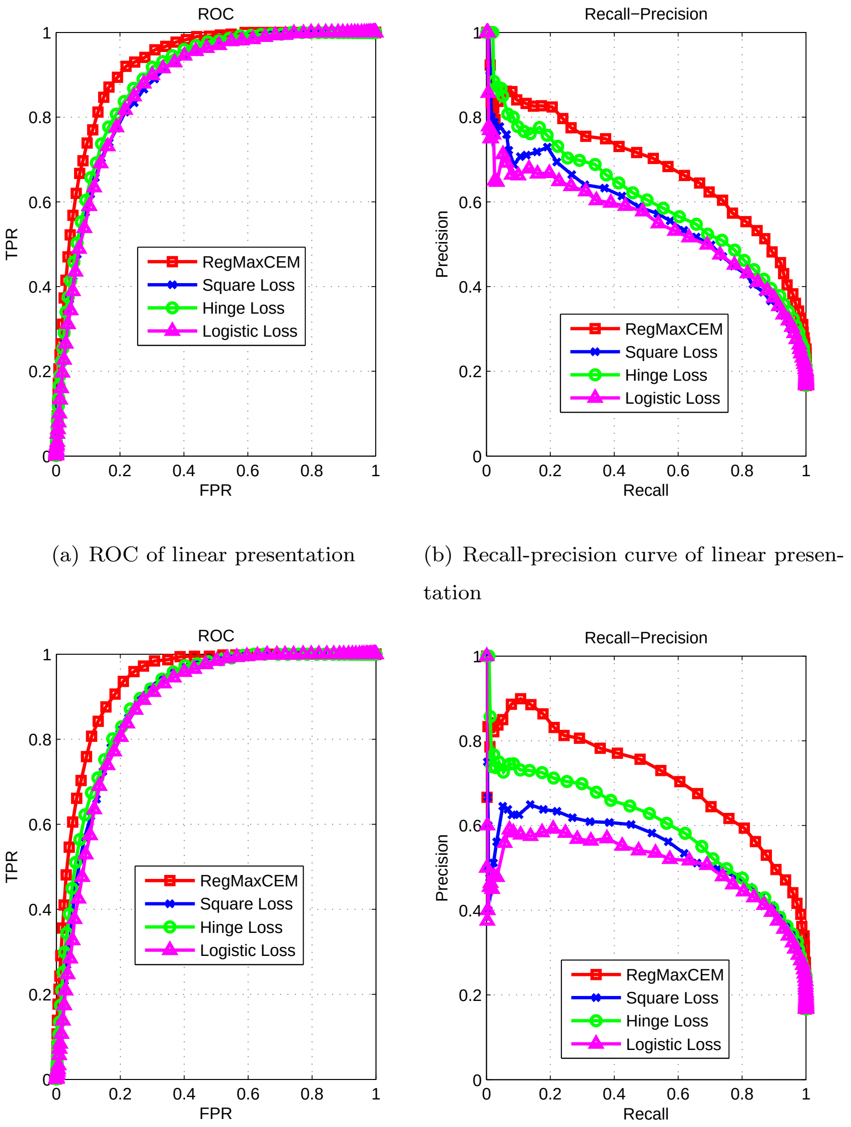
The prediction performance was measured by the receiver operating characteristic (ROC) and recall-precision curves. The usage of ROC curve is mainly due to the imbalanced classes. The ROC curve is created by plotting false positive rate (FPR) against true positive rate (TPR), while recall-precision curve is obtained by ploting recall against precision. The FPR, TPR, recall and precision are defined as:

$$\begin{aligned} FPR &= \frac{FP}{FP+TN},\ TPR = \frac{TP}{TP+FN}, \\ recall &= \frac{TP}{TP+FN},\ precision = \frac{TP}{TP+FP}, \end{aligned} \tag{19}$$

where TP is the number of DNA-binding sites predicted correctly, FP is the number of non-binding sites predicted as DNA-binding sites wrongly, TN is the number of non-binding sites predicted correctly, while FN is the number of DNA-binding sites predicted as non-binding sites wrongly. For a better predictor, its ROC curve should be closer to the top left corner of the figure, while the recall-precision curve should be closer to the top right corner. Besides the two curves, area under the ROC curve (AUC) is also used as a single measurement of the prediction. A better predictor will have a larger AUC value.

3.2.2. Results

The ROC and recall-precision curves of the proposed method and compared methods are reported in Figure 2. The predictors using linear and kernel representations are both illuminated. The AUC values of the ROC curves are reported in Table 3 as well. Overall the proposed methods clearly outperform the other methods significantly, although there is some variability in prediction performance over different representation types. From Table 3, we could see that the accuracy of the predictor is slightly increased by using the kernel representation instead of the linear representation. The regularized correntropy based predictors gives much better results than other methods on both representations. An interesting result from the DNA-binding prediction on this dataset is that the predictor with the hinge loss function outperforms other two methods.



(a) ROC of linear presentation

(b) Recall-precision curve of linear presentation

(c) ROC of kernel presentation

(d) Recall-precision curve of kernel presentation

Figure 2: ROC and recall-precision curves on DNA-Binding site prediction experiment using both linear and kernel representations.

Table 3: AUC values of ROC curves on DNA-Binding site prediction experiment.

Linear representation		Kernel representation	
Mehtods	AUC	Mehtods	AUC
RegMaxCEM	0.9226	RegMaxCEM	0.9344
Square Loss	0.8768	Square Loss	0.8891
Hinge Loss	0.8908	Hinge Loss	0.8961
Logistic Loss	0.8747	Logistic Loss	0.8776

4. Conclusion and Future Work

In this paper, we present a novel regularized predictor learning model for multi-class pattern recognition problems. The predictor is learned by maximizing the correntropy between the prediction results and the true class labels. By applying the MCC rule, we could treat different training samples differently, so that the noisy and outlying training samples have less impact on the learning of predictors. Compared with the existing predictor models with various loss functions, it is robust to the noisy and outlying training samples. The experiments on bacteria identification and DNA-binding site prediction show that a good predictor may benefit much from a well designed loss function based on MCC. The proposed method outperformed the predictor with other popularly used loss functions. In the future, we will investigate if the regularized maximum correntropy framework can be used to regularize ranking score learning [23, 24], data representation [25, 26, 27, 28, 29, 30, 31, 32, 33] Moreover, we also plan to extend the proposed regularized correntropy based classifier for wireless sensor network [34, 35, 36, 37, 38, 39, 40], computer vision [41, 42, 43, 44, 45, 46, 46, 47, 48], and computer network security [49, 50, 51, 52, 53, 54, 55].

References

[1] Z. Lu, P. Han, L. Wang, J.-R. Wen, Semantic sparse recoding of visual content for image applications, IEEE Transactions on Image Processing 24 (1).

[2] H. Li, G.-Q. Wu, X.-G. Hu, J. Zhang, L. Li, X. Wu, K-means clustering with bagging and mapreduce, in: System Sciences (HICSS), 2011 44th Hawaii International Conference on, IEEE, 2011, pp. 1–8.

[3] Z. Lu, L. Wang, Noise-robust semi-supervised learning via fast sparse coding, Pattern Recognition 48 (2) (2015) 605–612.

[4] H. Li, X. Wu, Z. Li, W. Ding, Online group feature selection from feature streams, in: Twenty-Seventh AAAI Conference on Artificial Intelligence, AAAI, 2013, pp. 1627–1628.

[5] L. Zhou, Z. Lu, H. Leung, L. Shang, Spatial temporal pyramid matching using temporal sparse representation for human motion retrieval, The Visual Computer 30 (6-8) (2014) 845–854.

[6] Y. Zhou, L. Li, H. Zhang, Adaptive learning of region-based plsa model for total scene annotation, arXiv preprint arXiv:1311.5590.

[7] Z. Lu, Y. Peng, Learning descriptive visual representation by semantic regularized matrix factorization, in: IJCAI, AAAI Press, 2013, pp. 1523–1529.

[8] Y. Zhou, L. Li, T. Zhao, H. Zhang, Region-based high-level semantics extraction with cedd, in: Network Infrastructure and Digital Content, 2010 2nd IEEE International Conference on, IEEE, 2010, pp. 404–408.

[9] Z. Lu, Y. Peng, Unified constraint propagation on multi-view data., in: AAAI, 2013.

[10] G. Zhou, Z. Lu, Y. Peng, ℓ_1-graph construction using structured sparsity, Neurocomputing 120 (2013) 441–452.

[11] R. He, W.-S. Zheng, B.-G. Hu, Maximum Correntropy Criterion for Robust Face Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (8) (2011) 1561–1576.

[12] Z. Lu, Y. Peng, Heterogeneous constraint propagation with constrained sparse representation, in: Proceedings of IEEE International Conference on Data Mining, IEEE Computer Society, 2012, pp. 1002–1007.

[13] L. Li, J. Yang, K. Zhao, Y. Xu, H. Zhang, Z. Fan, Graph regularized non-negative matrix factorization by maximizing correntropy, arXiv preprint arXiv:1405.2246.

[14] Z. Lu, Y. Peng, Exhaustive and efficient constraint propagation: A graph-based learning approach and its applications, International Journal of Computer Vision 103 (3) (2013) 306–325.

[15] R. He, W.-S. Zheng, B.-G. Hu, X.-W. Kong, A Regularized Correntropy Framework for Robust Pattern Recognition, Vol. 23, 2011, pp. 2074–2100.

[16] N.-H. Yang, M.-M. Huang, R. He, X.-K. Wang, Robust Semi-supervised Learning Algorithm Based on Maximum Correntropy Criterion, Journal of Software 23 (2012) 279–88.

[17] H. Xu, C. Caramanis, S. Mannor, Robustness and regularization of support vector machines, The Journal of Machine Learning Research 10 (2009) 1485–1510.

[18] X.-T. Yuan, B.-G. Hu, Robust feature extraction via information theoretic learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 1193–1200.

[19] Z. Lu, Y. Peng, Image annotation by semantic sparse recoding of visual content, in: Proceedings of the 20th ACM international conference on Multimedia, ACM, 2012, pp. 499–508.

[20] Z. Lu, L. Wang, Learning descriptive visual representation for image classification and annotation, Pattern Recognition 48 (2) (2015) 498–508.

[21] H. Pei, J. Li, M. Lv, J. Wang, J. Gao, J. Lu, Y. Li, Q. Huang, J. Hu, C. Fan, A Graphene-Based Sensor Array for High-Precision and Adaptive

Target Identification with Ensemble Aptamers, Journal of the American Chemical Society 134 (33) (2012) 13843–13849.

[22] T. Li, Q. Li, S. Liu, G. Fan, Y. Zuo, Y. Peng, Predna: accurate prediction of dna-binding sites in proteins by integrating sequence and geometric structure information, Bioinformatics 29 (6) (2013) 678–685.

[23] J. J.-Y. Wang, Y. Sun, X. Gao, Sparse structure regularized ranking, Multimedia Tools and Applications (2014) 1–20.

[24] Z. Lu, H. H. Ip, Generalized relevance models for automatic image annotation, in: Advances in Multimedia Information Processing-PCM 2009, Springer Berlin Heidelberg, 2009, pp. 245–255.

[25] Q. Sun, J. Lu, Y. Wu, H. Qiao, X. Huang, F. Hu, Non-informative hierarchical bayesian inference for non-negative matrix factorization, Signal Processing 108 (2015) 309–321.

[26] Z. Lu, Unsupervised image segmentation using an iterative entropy regularized likelihood learning algorithm, in: Advances in Neural Networks-ISNN 2006, Springer Berlin Heidelberg, 2006, pp. 492–497.

[27] Z. Lu, Y. Peng, J. Xiao, Unsupervised learning of finite mixtures using entropy regularization and its application to image segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[28] Z. Lu, Y. Peng, H. H.-S. Ip, Spectral learning of latent semantics for action recognition, in: IEEE International Conference onComputer Vision (ICCV), IEEE, 2011, pp. 1503–1510.

[29] Z. Lu, Y. Peng, Latent semantic learning with structured sparse representation for human action recognition, Pattern Recognition 46 (7) (2013) 1799–1809.

[30] Z. Lu, Y. Peng, Latent semantic learning by efficient sparse coding with hypergraph regularization., in: AAAI, 2011.

[31] Z. Lu, A regularized minimum cross-entropy algorithm on mixture of experts for curve detection, in: International Conference on Neural Networks and Brain, Vol. 2, IEEE, 2005, pp. 656–660.

[32] Q. Zhao, Z. Lu, H. H. Ip, Action recognition based on learnt motion semantic vocabulary, in: Advances in Multimedia Information Processing-PCM 2010, Springer Berlin Heidelberg, 2010, pp. 193–202.

[33] Z. Lu, X. Lu, Z. Ye, Unsupervised image categorization using constrained entropy-regularized likelihood learning with pairwise constraints, in: Advances in Neural Networks–ISNN 2007, Springer Berlin Heidelberg, 2007, pp. 1193–1200.

[34] S. Qingquan, H. Fei, Q. Hao, Context awareness emergence for distributed binary pyroelectric sensors, in: Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on, IEEE, 2010, pp. 162–167.

[35] Q. Sun, F. Hu, Y. Wu, X. Huang, Primate-inspired adaptive routing in intermittently connected mobile communication systems, Wireless Networks 1–16.

[36] Y. Wu, F. Hu, Q. Sun, K. Bao, M. Guo, A fast raptor codes decoding strategy for real-time communication systems, Network and Communication Technologies 2 (2) (2013) p29.

[37] Q. Sun, W. Yu, N. Kochurov, Q. Hao, F. Hu, A multi-agent-based intelligent sensor and actuator network design for smart house and home automation, Journal of Sensor and Actuator Networks 2 (3) (2013) 557–588.

[38] Q. Sun, P. Wu, Y. Wu, M. Guo, J. Lu, Unsupervised multi-level non-negative matrix factorization model: Binary data case, Journal of Information Security 3 (2012) 245.

[39] F. Hu, Q. Sun, Q. Hao, Neuro-disorder patient monitoring via gait sensor networks, Intelligent Sensor Networks: The Integration of Sensor Networks, Signal Processing and Machine Learning (2012) 181.

[40] Q. Sun, F. Hu, Q. Hao, Mobile targets scenario recognition via low-cost pyroelectric sensor network system: Towards an accurate context identification, Proc. 2011 IEEE Students Tech. Sym (2011) 1–5.

[41] Z. Lu, H. H. Ip, Spatial markov kernels for image categorization and annotation, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 41 (4) (2011) 976–989.

[42] Z. Lu, H. H. Ip, Y. Peng, Contextual kernel and spectral methods for learning the semantics of images, IEEE Transactions on Image Processing 20 (6) (2011) 1739–1750.

[43] Z. Lu, Y. Peng, H. H. Ip, Gaussian mixture learning via robust competitive agglomeration, Pattern Recognition Letters 31 (7) (2010) 539–547.

[44] Z. Lu, An iterative entropy regularized likelihood learning algorithm for cluster analysis with the number of clusters automatically detected, in: International Conference on Neural Networks and Brain, Vol. 2, IEEE, 2005, pp. 650–655.

[45] X. Lu, Z. Lu, A publishing framework for digitally augmented paper documents: towards cross-media information integration, in: Advances in Multimedia Information Processing-PCM 2006, Springer Berlin Heidelberg, 2006, pp. 494–501.

[46] Z. Lu, Y. Peng, A semi-supervised learning algorithm on gaussian mixture with automatic model selection, Neural Processing Letters 27 (1) (2008) 57–66.

[47] Z. Lu, H. H. Ip, Q. He, Context-based multi-label image annotation, in: Proceedings of the ACM International Conference on Image and Video Retrieval, ACM, 2009, p. 30.

[48] Z. Lu, H. H.-S. Ip, Image categorization with spatial mismatch kernels, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 397–404.

[49] Z. Zhan, M. Xu, S. Xu, A characterization of cybersecurity posture from network telescope data, in: Proceedings of The 6th International Conference on Trustworthy Systems, InTrust '14.

[50] Z. Zhan, M. Xu, S. Xu, Characterizing honeypot-captured cyber attacks: Statistical framework and case study, Information Forensics and Security, IEEE Transactions on 8 (11) (2013) 1775–1789.

[51] S. Xu, W. Lu, L. Xu, Z. Zhan, Adaptive epidemic dynamics in networks: Thresholds and control, ACM Trans. Auton. Adapt. Syst. 8 (4) (2014) 19:1–19:19.

[52] L. Xu, Z. Zhan, S. Xu, K. Ye, Cross-layer detection of malicious websites, in: CODASPY, 2013, pp. 141–152.

[53] L. Xu, Z. Zhan, S. Xu, K. Ye, An evasion and Counter-Evasion study in malicious websites detection, in: 2014 IEEE Conference on Communications and Network Security (CNS) (IEEE CNS 2014), San Francisco, USA, 2014.

[54] S. Xu, W. Lu, Z. Zhan, A stochastic model of multivirus dynamics, Dependable and Secure Computing, IEEE Transactions on 9 (1) (2012) 30–45.

[55] S. Xu, H. Qian, F. Wang, Z. Zhan, E. Bertino, R. Sandhu, Trustworthy information: concepts and mechanisms, in: Web-Age Information Management, Springer, 2010, pp. 398–404.